

FIG. 1

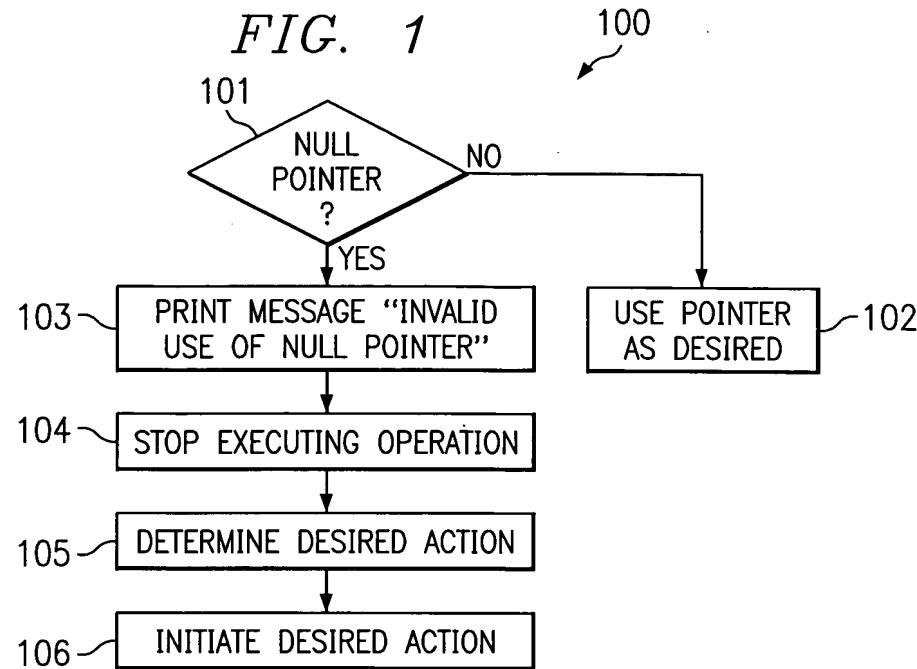


FIG. 2

```
01 ~ #include <iostream.h>
03 ~ template <class C>
04 ~ class SafePtr { 207      208
05 ~     public:           /
06 ~         SafePtr (C *p) : ptr(p) { }   / 210
07 ~         operator C *() const { return ptr; }
08 ~         C *operator-> () const {
09 ~             if (!ptr) {
10 ~                 cerr << "\nAttempted indirection of "
11 ~                     << "NULL pointer.\n";
12 ~                     exit(1); // or throw an exception
13 ~             }
14 ~             return ptr;
15 ~         }
16 ~         C operator* () const { return *operator->(); }
17 ~     private:
18 ~         C *const ptr;
19 ~     };

```

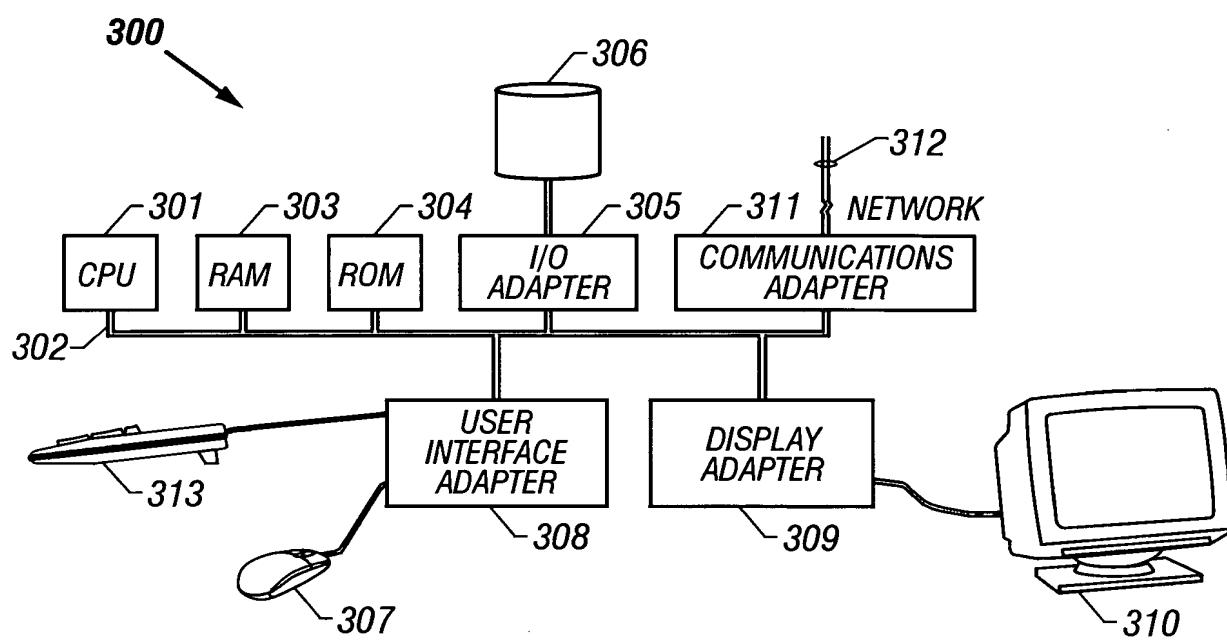


FIG. 3